LAB # 5

DECISION MAKING IN PROGRAMMING USING SWITCH STATEMENT

C++ Switch Statement

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case.

Syntax

The syntax for a switch statement in C++ is as follows −

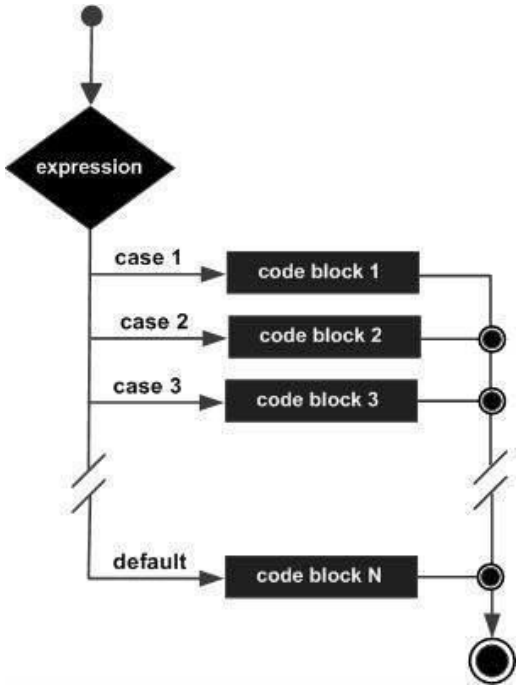
```
switch(expression)
{
        case constant-expression :
        statement(s);
        break; //optional

        case constant-expression :
        statement(s);
        break; //optional

        // you can have any number of case statements.
        default : //Optional
        statement(s);
}
```

The following rules apply to a switch statement −

- The expression used in a switch statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- The constant-expression for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.

- When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.
- A switch statement can have an optional default case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

Flow Diagram



Example

```
#include <iostream>
using namespace std;
int main ()
{
        // local variable declaration:
        char grade = 'B';
        switch(grade)
        {
                case 'A' :
                cout << "Excellent!" << endl;
                break;

                case 'B' :
```

```
            cout << "Good!" << endl;
            break;
            case 'C' :
            cout << "Well done" << endl;
            break;

            case 'D' :
            cout << "You passed" << endl;
            break;

            case 'F' :
            cout << "Better try again" << endl;
            break;

            default :
            cout << "Invalid grade" << endl;
      }
      cout << "Your grade is " << grade << endl;
      return 0;
}
```

This would produce the following result −

```
You passed
Your grade is B
```

goto Statement

goto statement can be used to branch towards any statement (forward or backward) which is marked by some label.

e.g.

```
statement 1;
here: statement 2;
….
….
goto here;
```

C++ Goto Statement Example

Let's see the simple example of goto statement in C++.

```
#include <iostream>
using namespace std;
int main()
```

```
{
ineligible:
      cout<<"You are not eligible to vote!\n";
    cout<<"Enter your age:\n";
    int age;
    cin>>age;
    if (age < 18){
          goto ineligible;
    }
    else
    {
          cout<<"You are eligible to vote!";
    }
}
```

Output:

```
You are not eligible to vote!
Enter your age:
16
You are not eligible to vote!
Enter your age:
7
You are not eligible to vote!
Enter your age:
22
You are eligible to vote!
```

Lab Tasks

1. What would be the output of following program?

a)
```
main( )
{
int i = 2 ;
        switch ( i )
        {
                case 1 :
                cout<<"I am in case 1 \n"<<endl;

                case 2 :
                cout<<"I am in case 2 \n"<<endl;

                case 3 :
                cout<<"I am in case 3 \n"<<endl;

                default :
                cout<<"I am in default \n"<<endl;
        }
}
```

2. Write a program to create Simple Calculator using switch case.

3. Write a program print total number of days in a month using switch case.

4. Develop a four basic mathematical functions calculator that takes input form user and performs any one of the selected operations and displays the result. Use switch case approach to develop this program. Also make use of "goto" statement to continuously take input of the operator, if user enters some unknown operator (other than, +,-, *, /)

5. Write a Program to take the value from the user as input all sides of a triangle and check whether the triangle is valid or not. Using switch statement

6. Using switch statement Write a program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade according to following:
   Percentage >= 90% : Grade A
   Percentage >= 80% : Grade B
   Percentage >= 70% : Grade C
   Percentage >= 60% : Grade D
   Percentage >= 40% : Grade E
   Percentage < 40% : Grade F